

means operative on the non-branch instructions in each said basic block for processing said instructions, and

means operative on said branch instruction in said basic block in response to the firing time information for completing the execution of said branch instruction during the same time as said processing means is processing the last to be executed non-branch instruction in said basic block so that the execution of said branch instruction occurs in parallel with the execution of said non-branch instructions in said basic block thereby speeding the overall processing of said program by said system.

E1
72. (Twice Amended) A system for executing branches in single entry-single exit (SESE) basic blocks (BBs) in a plurality of programs utilized by a number of users, each basic block having a plurality of non-branch instructions and a branch instruction, said system comprising:

means receptive of each said programs for determining the branch instruction within each said basic block of each of said programs, said determining means further adding firing time information to said branch instructions,

means operative on the non-branch instructions in each said basic block of each said program for processing said programs, and

means operative on said branch instructions in each said basic block in response to the firing time information for completing the execution of said branch instruction during the

E1
same time as said processing means is processing the last to be executed non-branch instruction in said basic block for a given program so that the execution of said branch instruction occurs in parallel with the execution of said non-branch instructions in said basic block thereby speeding up the overall processing all of said programs by said system.

74. (Twice Amended) A system for executing branches in single entry-single exit (SESE) basic blocks (BBs) contained within a program, said basic block having a plurality of non-branch instructions and a branch instruction, said system comprising:

E2
means receptive of said program for determining the branch instruction within each said basic block of said program, said determining means further scheduling processing of said branch instruction,

means operative on the non-branch instructions in each said basic block for processing said instructions, and

means operative on said branch instruction in said basic block for completing the execution of said scheduled branch instruction during an instruction cycle no later than (during) the same [time] instruction cycle as said processing means is processing the last to be executed non-branch instruction in said basic block so that the execution of said branch instruction occurs in parallel with the execution of said non-branch

instructions in said basic block thereby speeding up the overall processing of said program by said system.

E² 75. (Twice Amended) A system for executing branches in single entry-single exit (SESE) basic blocks (BBs) in a plurality of programs utilized by a number of users, said basic block having a plurality of non-branch instructions and a branch instruction, said system comprising:

means receptive of each said programs for determining the branch instruction within each said basic block of each of said programs, said determining means further scheduling processing of said branch instructions,

means operative on the non-branch instructions in each said basic block of each said program for processing said programs, and

means operative on said branch instruction in each said basic block for completing the execution of said scheduled branch instruction during an instruction cycle no later than during the same [time] instruction cycle was said processing means is processing the last to be executed non-branch instruction in said basic block for a given program so that the execution of said branch instruction occurs in parallel with the execution of said non-branch instructions in said basic block whereby overall processing throughput of all said programs by said system is increased.

76. (Twice Amended) A system for executing scheduled branches in single entry-single exit (SESE) basic blocks (BBs) contained within a program, each basic block having a plurality of non-branch instructions and a branch instruction, said system comprising:

E2 means receptive of said program for determining the branch instruction within each said basic block of said program said determining means further adding instruction firing time information to said scheduled branch instruction,

means operative on the non-branch instructions in each said basic block for processing said non-branch instructions, and

means operative on said branch instruction in said basic block in response to said time information, for completing the execution of said scheduled branch instruction during the same time as said processing means is processing the last to be executed non-branch instruction in said basic block so that the execution of said branch instruction occurs in parallel with the execution of said non-branch instructions in said basic block thereby speeding up the overall processing of said program by said system.

77. (Twice Amended) A system for executing scheduled branches in single entry-single exit (SESE) basic blocks (BBs) in a plurality of programs utilized by a number of users, each basic block having a plurality of non-branch instructions and a branch instruction, said system comprising:

means receptive of each said programs for determining the branch instruction within each said basic block of each of said programs, said determining means further adding instruction firing time information to said scheduled branch instructions,

means operative on the non-branch instructions in each said basic block of each said program for processing said programs, and

means operative on said branch instructions in each said basic block for completing the execution of said scheduled branch instruction during the same time as said processing means is processing the last to be executed non-branch instruction in said basic block for a given program so that the execution of said branch instruction occurs in parallel with the execution of said instructions in said basic block whereby overall processing throughput of all said programs by said system is increased.

82

78. (Twice Amended) A method for operating a programmed computer for executing branches in single entry-single exit (SESE) basic blocks (BBs) contained within a program, each said basic block having a plurality of non-branch instructions and a branch instruction, said method comprising the steps of:

determining the branch instruction within each said basic block of said program,

adding information to said branch instruction,

processing said instructions in each said basic block, and

completing the execution of said branch instruction in said basic block, based upon said added information, during an

E2
instruction cycle no later than during the [time duration]
instruction cycle for [of] processing the last to be executed non-branch instruction in said basic block so that the execution of said branch instruction occurs in parallel with the execution of said non-branch instructions in said basic block thereby speeding up the overall processing of said program.

80. (Twice Amended) A method for operating a programmed computer for executing branches in single entry-single exit (SESE) basic blocks (BBs) contained within a program, each basic block having a plurality of non-branch instructions and a branch instruction, said method comprising the steps of:

E3
determining the branch instruction within each said basic block of said program,
scheduling processing of said branch instruction,
processing said instructions in each said basic block, and
completing the execution of said scheduled branch instruction during an instruction cycle no later than during the processing of the last to be executed non-branch instruction in said basic block so that the execution of said branch instruction occurs in parallel with the execution of said non-branch instructions in said basic block thereby speeding up the overall processing of said program.

81. (Twice Amended) A method for operating a programmed computer for executing branches in single entry-single exit (SESE)

basic blocks (BBs) in a plurality of programs utilized by a number of users, each basic block having a plurality of non-branch instructions and a branch instruction, said method comprising the steps of:

determining the branch instruction within each said basic block of each said programs,

scheduling processing of said branch instructions,

processing the instructions in each said basic block of each said program, and

E³ completing the execution of said scheduled branch instruction during an instruction cycle occurring no later than during the processing of the last to be executed non-branch instruction in said basic block for a given program so that the execution of said branch instruction occurs in parallel with the execution of said non-branch instructions in said basic block whereby overall processing throughput of all said programs is increased.

82. (Twice Amended) A method for operating a programmed computer of executing scheduled branches in single entry-single exit (SESE) basic blocks (BBs) contained within a program, each basic block having a plurality of non-branch instructions and a branch instruction, said method comprising the steps of:

determining the branch instruction within each said basic block of said program,

adding instruction firing time information to said branch instruction for scheduling said branch instruction,

processing said instructions in each said basic block, and completing the execution of said scheduled branch instruction according to said firing time information during an instruction cycle occurring no later than during the processing of the last to be executed non-branch instruction in said basic block so that the execution of said branch instruction occurs in parallel with the execution of said non-branch instructions in said basic block thereby speeding up the overall processing of said program.

E3 83. (Twice Amended) A method for operating a programmed computer for executing scheduled branches in single entry-single exit (SESE) basic blocks (BBs) in a plurality of programs utilized by a number of users, each basic block having a plurality of non-branch instructions and a branch instruction, said method comprising the steps of:

determining the branch instruction within each said basic block of each of said programs,

adding instruction firing time information to said scheduled branch instruction for scheduling processing of said branch instruction,

processing the instructions in each said basic block of said programs, and

completing the execution of said scheduled branch instruction according to said firing time information during an instruction cycle occurring no later than during the processing of the last to be executed non-branch instruction in said basic block for a given

program so that the execution of said branch instruction occurs in parallel with the execution of said non-branch instructions in said basic block whereby overall processing of all said programs is increased.

[Add claim 84 as follows:]

84. A method for operating a programmed computer for executing branches in single entry-single exit (SESE) basic blocks (BBs) in a plurality of programs utilized by a number of users, each basic block having a stream of instructions including a plurality of non-branch instructions and a branch instruction, said method comprising the steps of:

determining the branch instruction within each said basic block of each of said programs,

adding information to said branch instructions,

processing the instruction in each said basic block of each said program, and

completing the execution of said branch instructions in each said basic block based upon said added information, during an instruction cycle occurring no later than during the instruction cycle for processing the last to be executed non-branch instruction in each respective basic block for a given program so that the execution of said branch instruction occurs in parallel with the execution of said non-branch instructions in said basic block thereby speeding up the overall processing of said programs.